

<! – Program Number 1a .Develop and demonstrate a XHTML file that includes -----

Javascript script for the following problems:--

a) Input: A number n obtained using prompt

Output: The first n Fibonacci numbers -->

<!-- Program Name fib.html-->

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html PUBLIC "-//w3c//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Fibonacci Numbers</title>
<script>
function fib(num)
{
    if (num == 0)
    {
        var fibnum = 0;
    }
    else
    {
        if (num == 1)
        {
            fibnum = 1;
        }
        else
        {
            fibnum = fib(num - 2) + fib(num - 1);
        }
    }
    return fibnum;
}

function wf()
{
    var num = prompt("Enter Number","");
    num = parseInt(num);
    document.write ("Fibanocci Series Are <br>");
    for (var i=0; i <= num ; i++)
    {
```

```
        document.write (fib(i) + " <br>");
    }
}
</script>    </head>
<body onload="wf()">
</body>
</html>
```

<!-- Program Number 1b

--Input: A number n obtained using prompt --

-- Output: A table of numbers from 1 to n and their squares using alert --

-- Prgram Name sqr.html -->

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html PUBLIC "-//w3c//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Square</title>
<script>
function sqr()
{
var num = prompt("Enter Number","");
num = parseInt(num);
document.writeln("<table border='1'>");
document.writeln("<caption>Numbers and Squares</caption>");
document.writeln("<tr>");
document.writeln("<th>Number</th>");
document.writeln("<th>Square</th>");
document.writeln("</tr>");
    for (var i=0; i <= num ; i++)
    {
        var square=i*i;
        alert(i+"---->" +i*i);
        document.writeln("<tr>");
        document.writeln("<td>" +i+"</td>");
        document.writeln("<td>" +square+"</td>");
        document.writeln("</tr>");
    }
}
```

```
    }
    document.writeln("</table>");
  }
</script>
</head>
<body onload="sqr()">
</body>
</html>
```

<! -- Program Number 2a --

Develop and demonstrate, using Javascript script, a XHTML document that collects the USN (the valid format is: A digit from 1 to 4 followed by two upper case characters followed by two digits followed by two upper-case characters followed by three digits; no embedded spaces allowed) of the user. Event handler must be included --for the form element that collects this information to validate the input. Messages in the alert windows must be produced when errors are detected. --

Prgram Name val usn.html -->

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html PUBLIC "-//w3c//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Validate USN</title>
<script>
function val(usn)
{
var ok=usn.search(/[1-4][A-Z]{2}\d{2}[A-Z]{2}\d{2}[1-9]/);
var usn1=document.getElementById("usn").value;
    if(ok==0 && usn.length==10)
    {
        alert("The USN "+usn1+" is Valid");
    }

    else
    {
        alert("The USN "+usn1+" is Invalid");
        alert("plz enter in DUUDDUUDDD Format");
    }
}
```

```
}
</script>
</head>
<body>
<form >
<br>
Enter USN
<br>
<input type=text name="usn" ID="usn">
<input type=button name="btnVal" ID="btnVal" value="Validate" onclick='val(usn.value);'>
</form>
</body>
</html>
```

OR

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html PUBLIC "-//w3c//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Validate USN</title>
<script>
function val(usn)
{
var ok=usn.search(/[1-4][A-Z]{2}\d{2}[A-Z]{2}\d{3}/);
var usn1=document.getElementById("usn").value;
var first=usn1.charAt(7)+usn1.charAt(8)+usn1.charAt(9);
    if(first!=000 && ok==0 && usn.length==10)
    {
        alert("The USN "+usn1+" is Valid");
    }
    else
    {
        if(first==000)
        {
            alert("The USN "+usn1+" is Invalid");
            alert("plz Do Not Enter ZERO as USN");
            alert("plz enter in DUUDDUDDDD Format");
        }
    }
}
```

```
        }
        else
        {
            alert("The USN "+usn1+" is InValid");
            alert("plz enter in DUUDDUUDDD Format");
        }
    }
}
</script>
</head>
<body>
<form name="formusn" ID="formusn">
<br>
Enter USN
<br>
<input type="text" name="usn" ID="usn">
<input type="button" name="btnVal" ID="btnVal" value="Validate" onclick='val(usn.value);'>
</form>
</body>
</html>
```

<!-- Program Number 2b --

Develop and demonstrate, using Javascript script, a XHTML document that collects the the current semester (restricted to be a number from 1 to 8)Event handler must be included for the form element that collects this information to validate the input. Messages in the alert windows must be produced when errors are detected. Program Name valsem.html -->

```
<?xml version = "1.0" encoding = "utf-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns = "http://www.w3.org/1999/xhtml">
<head>      <title>Program 4b</title>
</head>
<script type="text/javascript">
    function validate()
    {
        var pattern1=/([1-4][A-Z]{2}\d{2}[A-Z]{2}\d{2}[1-9])/;
        var pattern2=/[1-8]/;
        usn = document.getElementById('usn');
        sem = document.getElementById('sem');
        if( usn.value.match(pattern1) )
            alert("The Entered" +usn.value+ "valid USN");
        else
            alert("The Entered" +usn.value+ "Invalid USN");
        if( sem.value.match(pattern2) &&sem.value.length==1)
            alert("The Entered" +sem.value+ "Valid SEM");
        else
            alert("The Entered" +sem.value+ "Invalid SEM");
    }
</script>
<body>
<form>
Enter USN: <input type="text" name="usn" id="usn" />
Enter SEM: <input type="text" name="sem" id="sem" />
<input type="submit" value="Check" onclick="validate()"/>
</form>
</body>
```

</html>

Program 3a

Develop and demonstrate, using JavaScript, a XHTML document that contains three short paragraphs of text, stacked on the top of each other, with only enough of each showing so that the mouse cursor can be placed over some part of them. When the cursor is placed over the exposed part of any paragraph, it should rise to the top to completely visible.

< -- pgm3a.html -- >

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<!DOCTYPE html PUBLIC "-//w3c//DTD XHTML 1.1//EN"
```

```
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
  <head>
```

```
    <title>
```

The Stacking of paragraphs when moved from top stack position, returns to its original position.

```
  </title>
```

```
    <style type="text/css">
```

```
      .para1
```

```
      {
```

```
          border: solid thick #c0c0c0;
```

```
          padding: 1in;
```

```
          width:180px;
```

```
          background-color:#800000;
```

```
          position:absolute;
```

```
          top:70px;
```

```
          left:5in;
```

```
          z-index:1;
```

```
      }
```

```
      .para2
```

```
      {
```

```
          border: solid thick #808000;
```

```
          padding: 1in;
```

```
          width:180px;
```

```
          background-color:#000080;
```

```
          position:absolute;
```

```
          top:105px;
```

```
          left:5.5in;
```

```
          z-index:2;
```

```
    }
    .para3
    {
        border: solid thick #00ffff;
        padding: 1in;
        width:180px;
        background-color:#ff00ff;
        position:absolute;
        top:140px;
        left:6in;
        z-index:3;
    }
    .display
    {
        font-size:25pt;
        color:lime;
        text-align:center;
    }
    p:hover{background-color:rgb(195,217,255);font-size:25px;
};

</style>
<script type="text/javascript">
    var stack1="stack3";
    function move(curStack)
    {
        var newStack=document.getElementById(curStack).style;
        var oldStack=document.getElementById(stack1).style;
        oldStack.zIndex="0";
        newStack.zIndex="10";
        stack1=curStack;
    }
</script>
</head>
<body>
<h2 class="display">Stacking of Paragraphs on top of each other</h2>
<p class="para1" id="stack1" onmouseover="move('stack1')">
    Welcome to Programming the WEB.
</p>
<p class="para2" id="stack2" onmouseover="move('stack2')">
```



```
Welcome to RNSIT, Bengaluru!  
</p>  
<p class="para3" id="stack3" onmouseover="move('stack3')">  
  Welcome to CSE DEPT, RNSIT, Bengaluru!  
</p>  
</body>  
</html>
```

Program 3b

Modify the above document so that when a paragraph is moved from top stacking position, it returns to its original position rather than the bottom

```
< -- pgm5b.html -- >  
<?xml version="1.0" encoding="utf-8" ?>  
<!DOCTYPE html PUBLIC "-//w3c//DTD XHTML 1.1//EN"  
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
  <title>Stack to original position</title>  
  <style type="text/css">  
    .para1  
    {  
      border: solid thick #c0c0c0;  
      padding: 1in;  
      width:180px;  
      background-color:#800000;  
      position:absolute;  
      top:100px;  
      left:5in;  
      z-index:1;  
    }  
    .para2  
    {  
      border: solid thick #808000;  
      padding: 1in;  
      width:180px;  
      background-color:#000080;
```

```
        position:absolute;
        top:135px;
        left:5.5in;
        z-index:2;
    }
    .para3
    {
        border: solid thick #00ffff;
        padding: 1in;
        width:180px;
        background-color:#ff00ff;
        position:absolute;
        top:170px;
        left:6in;
        z-index:3;
    }
    .display
    {
        font-size:25pt;
        color:lime;
        text-align:center;
    }
    p:hover{background-color:rgb(195,217,255);font-size:25px;};
</style>
<script type="text/javascript">
    var stack1;
        var origpos;
    function move(curStack,pos) {
        var newStack=document.getElementById(curStack).style;
        newStack.zIndex="10";
        stack1=curStack;
        origpos=pos;
    }
    function originalPosition(){
        document.getElementById(stack1).style.zIndex=origpos;
    }
</script>
</head>
```

```
<body>
  <h2 class="display">The Stacking of paragraphs when moved from top stack position
returns to its original position.</h2>
  <p class="para1" id="uniq1" onmouseover="move('uniq1',1)"
onmouseout="originalPosition()">
    Welcome to Programming the WEB.
  </p>
  <p class="para2" id="uniq2" onmouseover="move('uniq2',2)"
onmouseout="originalPosition()">
    Welcome to RNSIT Bangalore!
  </p>
  <p class="para3" id="uniq3" onmouseover="move('uniq3',3)"
onmouseout="originalPosition()">
    Welcome to CSE DEPT, RNSIT, Bengaluru!
  </p>
</body>
</html>
```

<!-- Program 4a-->

Design an XML document to store information about a student in an engineering college affiliated to VTU. The information must include USN, Name, Name of the College, Brach, Year of Joining, and e-mail id. Make up sample data for 3 students. Create a CSS style sheet and use it to display the document. Program Name stu.xml.

```
<?xml version="1.0" encoding="utf-8" ?>
<?xml-stylesheet type="text/css" href="pgm6a.css" ?>
<html>
<head>
<h1> STUDENTS DESCRIPTION </h1>
</head>
<students>
<student>
  <USN> USN: 1RN07CS001</USN>
  <name> NAME: Anjan</name>
  <college>COLLEGE: RNSIT</college>
  <branch> BRANCH: Computer Science and Engineering</branch>
```

```
<year> YEAR:          2007</year>
<e-mail> E-Mail:      anjan@gmail.com</e-mail>
</student>
<student>
  <USN>USN:           1RN07IS001</USN>
  <name>NAME:         Anil</name>
  <college>COLLEGE:   RNSIT</college>
  <branch>BRANCH:     Information Science and Engineering</branch>
  <year>YEAR:         2007</year>
  <e-mail>E-Mail      anil@gmail.com</e-mail>
</student>
<student>
  <USN>USN:           1RN07EC001</USN>
  <name>NAME:         Avinash</name>
  <college>COLLEGE:   RNSIT</college>
  <branch>BRANCH:     Electronics and Communication Engineering</branch>
  <year>YEAR:         2007</year>
  <e-mail>E-Mail:     avinash@gmail.com</e-mail>
</student>
</students>
</html>
```

<!-- Program number 4a CSS-->

<!-- program name pgm4a.css-->

```
student
{
    display:block; margin-top:10px; color:Navy;
}
USN
{
    display:block; margin-left:10px;font-size:14pt; color:Red;
}
name
{
    display:block; margin-left:20px;font-size:14pt; color:Blue;
}
college
{
    display:block; margin-left:20px;font-size:12pt; color:Maroon;
```

```
}  
branch  
{  
    display:block; margin-left:20px;font-size:12pt; color:Purple;  
}  
year  
{  
    display:block; margin-left:20px;font-size:14pt; color:Green;  
}  
e-mail  
{  
    display:block; margin-left:20px;font-size:12pt; color:Blue;  
}
```

<!-- Program 4b-->

Design an XML document to store information about a student in an engineering college affiliated to VTU. The information must include USN, Name, Name of the College, Branch, Year of Joining, and e-mail id. for 1 student Create an XSLT style sheet for one student element.

<!-- Program Name pgm4b.xml-->

```
<?xml version="1.0" encoding="utf-8" ?>  
<?xml-stylesheet type="text/xsl" href="Student.xsl" ?>  
<student>  
  <USN>1RN07CS001</USN>  
  <name>Anil</name>  
  <college>RNSIT</college>  
  <branch>Computer Science and Engineering</branch>  
  <year>2007</year>  
  <e-mail>anjankn05@gmail.com</e-mail>  
</student>
```

<!-- Prgram 4b XSLT Style Sheet document Create an XSLT style sheet for one student element-->

<!-- pgm4b.xsl-->

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns="http://www.w3.org/1999/xhtml">
<xsl:template match="/student">
<html>
<head>
<title>Style sheet for StudentXsl.xml</title>
</head>
<body>
<h2>Student Details</h2>
<span style="font-style:italic;color:blue;">USN:</span>
<xsl:value-of select="USN"/><br/>
<span style="font-style:italic;color:blue;">Name:</span>
<xsl:value-of select="name"/><br/>
<span style="font-style:italic;color:blue;">College:</span>
<xsl:value-of select="college"/><br/>
<span style="font-style:italic;color:blue;">Branch:</span>
<xsl:value-of select="branch"/><br/>
<span style="font-style:italic;color:blue;">Year:</span>
<xsl:value-of select="year"/><br/>
<span style="font-style:italic;color:blue;">e-mail:</span>
<xsl:value-of select="e-mail"/><br/>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

5a. Write a Perl program to display various Server Information like Server Name, Server Software, Server protocol, CGI Revision etc.

```
/* PGM 5a.pl/
#!/usr/bin/perl
use CGI:standard;
print "content-type:text/html","\n\n";
print "<html>\n";
print "<head> <title> About this server </title> </head>\n";
print "<body><h1> About this server </h1>","\n";
print "<hr>";
print "Server name :",$ENV{'SERVER_NAME'},"<br>";
print "Running on port :",$ENV{'SERVER_PORT'},"<br>";
print "Server Software :",$ENV{'SERVER_SOFTWARE'},"<br>";
print "CGI-Revision :",$ENV{'GATEWAY_INTERFACE'},"<br>";
print "<hr>\n";
print "</body></html>\n";
exit(0);
```

5b. Write a Perl program to accept UNIX command from a HTML form and to display the output of the command executed.

```
/* PGM 5b.pl/
#!/usr/bin/perl
use CGI:standard;
print "content-type: text/html \n\n";
$c=param('com');
system($c);
exit(0);

/* PGM 5b.html/
<html>
<body>
<form action="http://localhost/cgi-bin/p5b.pl">
<input type="text" name="com">
<input type="submit" value="Submit">
</body>
</form>
</html>
```

6a. Write a Perl program to accept the User Name and display a greeting message randomly chosen from a list of 4 greeting messages.

/ PGM 6a.pl/*

```
#!/usr/bin/perl
use CGI ':standard';
use CGI::Carp qw(warningsToBrowser);
@coins = ("Welcome to Web Programming Lab", "Have a nice time in lab", "Practice all the
programs", "well done Good Day");
$range = 4;
$random_number = int(rand($range));
if(param)
{
print header();
print start_html(-title=>"User Name",-bgcolor=>"Pink",-text=>"blue");
$cmd=param("name");
print b("Hello $cmd, $coins[$random_number]"),br();
print start_form();
print submit(-value=>"Back");
print end_form();
print end_html();
}
else
{
print header();
print start_html(-title=>"Enter user name",-bgcolor=>"yellow",-text=>"blue");
print start_form(),textfield(-name=>"name",-value=>" "), submit(-name=>"submit",-
value=>"Submit"),reset();
print end_form();
print end_html();
}
```


6b. Write a Perl program to keep track of the number of visitors visiting the web page and to display this count of visitors, with proper headings.

/ PGM 6b.pl/*

```
#!/usr/bin/perl
print "content-type:text/html","\n\n";
use CGI ':standard';
$cnt="/var/www/html/c.txt";
print "welcome to our page", "<br/>";
print "nummber of times visited is", "<br/>";
if(open(FILE,"<".$cnt))
{
    $no_ac=<FILE>;
    close(FILE);
    if(open(DATA,">".$cnt))
    {
        $no_ac++;
        print DATA $no_ac;
        close(DATA);
        print $no_ac;
    }
    else
    {
        print "cant open a file for writing";
    }
}

else
{
    print "cant open file for read";
}
```

7. Write a Perl program to display a digital clock which displays the current time of the server.

```
/* PGM 7.pl/
#!/usr/bin/perl
use CGI ':standard';
print "Refresh: 1\n";
print "content-type: text/html\n\n";
print start_html(-title=>"Program 8",-bgcolor=>"Black",-text=>"white");
($s,$m,$h)=localtime(time);
print br,br,"The current system time is $h:$m:$s";
print br,br,hr,"In words $h hours $m minutes $s seconds";
print end_html;
```

8. Write a Perl program to insert name and age information entered by the user into a table created using MySQL and to display the current contents of this table.

i. Goto mysql then

Create database anjan;

use anjan;

Create table stud(name varchar(25),age varchar(25),);

/PGM 8.html/

```
<html>
```

```
<body>
```

```
<form action="http://localhost/cgi-bin/p10p.pl">
```

```
  Name : <input type="text" name="name"> <br>
```

```
  Age : <input type="text" name="age"> <br>
```

```
<input type="submit" value="Submit">
```

```
</form>
```

```
</html>
```

/PGM 8.pl/

```
#!/usr/bin/perl
```

```
print "Content-type: text/html\n\n";
```

```
print "<html><title>Result of the insert operation </title>";
```

```
use CGI ':standard';
```

```
use DBI;
```

```
$dbh=DBI->connect("DBI:mysql:anjan","root","");
```

```
$name=param("name");
```

```
$age=param("age");
```

```
$qh=$dbh->prepare("insert into stud values('$name','$age')");
```

```
$qh->execute();
```

```
$qh=$dbh->prepare("Select * from stud");
```

```
$qh->execute();
```

```
print "<table border size=1><tr><th>Name</th><th>Age</th></tr>";
```

```
while ( ($name,$age)=$qh->fetchrow())
```

```
{
```

```
    print "<tr><td>$name</td><td>$age</td></tr>";
```

```
}
```

```
print "</table>";
```

```
$qh->finish();
```

```
$dbh->disconnect();
```

```
print"</html>";
```

9. Write a PHP program to store current date-time in a COOKIE and display the 'Last visited on' date-time on the web page upon reopening of the same page.

/PGM p9.php/

```
<?php
    $Month = 2592000 + time();
    //this adds 30 days to the current time
    setcookie(AboutVisit, date("F jS - g:i a"), $Month);
//this creates cookie
?>
<html>
    <body>
        <?php
            //Below checks cookie existence
            if(isset($_COOKIE['AboutVisit']))
            {
                $last = $_COOKIE['AboutVisit'];
                echo "Welcome back! <br> You last visited on ". $last;
            }
            else
            {
                echo "Welcome to our site!";
            }
        ?>
    </body>
</html>
```

10. Write a PHP program to store page views count in SESSION, to increment the count on each refresh, and to show the count on web page.

/*Session.php*/

```
<?php
    session_start();
    if(isset($_SESSION['views']))
        $_SESSION['views']=$_SESSION['views']+1;
    else
        $_SESSION['views']=1;
    //echo "Views=". $_SESSION['views'];
?>
<html>
    <body>
        <?php
            //retrieve session data
            echo "Pageviews=". $_SESSION['views'];
        ?>
    </body>
</html>
```

11. Create a XHTML form with Name, Address Line 1, Address Line 2, and E-mail text fields. On submitting, store the values in MySQL table. Retrieve and display the data based on Name.

MySQL Table:

```
CREATE TABLE Persons
(
  name varchar(25),
  address1 varchar(25),
  address2 varchar(25),
  email varchar(25)
)
```

Program Code:

pg11.html

```
<html>
  <body>
    <form action="/cgi-bin/pg13in.php" method="post">
      Name: <input type="text" name="name" />
      Address Line 1: <input type="text" name="address1" />
      Address Line 2: <input type="text" name="address2" />
      Email: <input type="text" name="email" />
      <input type="submit" />
    </form>
  </body>
</html>
```

2. pg11in.php

```
<html>
  <body>
    <?php
      $con = mysql_connect("localhost","root","");
      if (!$con)
      {
        die('Could not connect: ' . mysql_error());
      }
      mysql_select_db("anjan");
      $sql="INSERT INTO Persons (name, address1, address2, email)
VALUES ('$_POST[name]','$_POST[address1]','$_POST[address2]','$_POST[email]')";

      if (!mysql_query($sql,$con))
```

```
{
    die('Error: ' . mysql_error());
}

echo "1 record added";
mysql_close($con)
?>

<hr>
<hr>
<div>
<h2><font color="red"><marquee><blink> Enter name u want to
search</blink></marquee></font></h2>
<hr>
    <form action="pg13re.php" method="post">
        Name: <input type="text" name="name" />
        <input type="submit" value="Submit" />
        <input type="reset" value="Clear"/>
    </form>
</div>
</body>
</html>
```

3.pg11 re.php

```
<html>
    <body>
        <?php
            $con = mysql_connect("localhost","root","");
            if (!$con)
            {
                die('Could not connect: ' . mysql_error());
            }

            mysql_select_db("anjan");

            $result = mysql_query("SELECT * FROM Persons where name=
'$_POST[name]'");

            if(!$result)
            {
                echo "1 record added";
            }
        }
    </body>
</html>
```

```
    }
    echo "<table border='1'>
    <tr>
    <th>Name</th>
    <th>Addresss1</th>
    <th>Addresss2</th>
    <th>Email</th>
    </tr>";

    while($row = mysql_fetch_array($result))
    {
        echo "<tr>";
        echo "<td>" . $row['name'] . "</td>";
        echo "<td>" . $row['address1'] . "</td>";
        echo "<td>" . $row['address2'] . "</td>";
        echo "<td>" . $row['email'] . "</td>";
        echo "</tr>";
    }
    echo "</table>";

    mysql_close($con);
?>
</body>
</html>
```

3. pg11re.php

```
<html>
    <body>
        <?php
            $con = mysql_connect("localhost","root","");
            if (!$con)
            {
                die('Could not connect: ' . mysql_error());
            }
            mysql_select_db("anjan");
            $result = mysql_query("SELECT * FROM Persons where name= '$_POST[name]'");
            if ($result == FALSE)
                echo "NO";
```



```
$Rows=mysql_num_rows($result);
if($Rows==0)
{
    echo $result;
    echo "No record is Found...";
}
else
{
    echo "<table border='1'>
    <tr>
    <th>Name</th>
    <th>Addresss1</th>
    <th>Addresss2</th>
    <th>Email</th>
    </tr>";
    while($row = mysql_fetch_array($result))
    {
        echo "<tr>";
        echo "<td>" . $row['name'] . "</td>";
        echo "<td>" . $row['address1'] . "</td>";
        echo "<td>" . $row['address2'] . "</td>";
        echo "<td>" . $row['email'] . "</td>";
        echo "</tr>";
    }
    echo "</table>";
}
mysql_close($con);
?>
</body>
</html>
```

12. Build a Rail's application to accept Book information viz accession number, title author's , edition and publisher from a web page and store the information in a data base and to search for a with title specified by the user and to display the search results with proper headings.

Creating Books Database

mysql -u root

create database lab12_development;

Web Programming Lab Manual

```
create database lab12_test;
```

```
create database lab12_production;
```

```
use lab12_development;
```

```
create table books (
```

```
id int not null auto_increment,
```

```
name varchar(80) not null,
```

```
description text not null,
```

```
price decimal(8, 2) not null,
```

```
primary key(id)
```

```
);
```

Creating lab12 Project

```
rails -d mysql lab12
```

Creating controller, model and view from database

```
ruby script/generate scaffold Book name:string description:text price:float
```

Starting Rails Server

```
ruby script/server
```

Executing in the web browser

<http://localhost:3000/books>

Creating main (new) controller for searching book

```
ruby script/generate controller main
```

Opening main controller program

Web Programming Lab Manual

edit app\controllers\main_controller.rb

Create two views welcome and result

```
class MainController < ApplicationController
```

```
  def welcome
```

```
    @num_books = Book.count
```

```
  end
```

```
  def result
```

```
    @bookid = params[:sid]
```

```
    @bookz = Book.find(:all, :conditions => "id = #{@bookid}")
```

```
  end
```

```
end
```

Create Result view file

> notepad app\views\main\result.rhtml

```
<html>
```

```
<title> Welcome template for books </title>
```

```
<body>
```

```
<p> Entered book id is <%= @bookid %> </p>
```

```
<table border=1>
```

```
<tr><th>Book Id</th><th>Book Name</th><th>Details </th> <th>Price </th> </tr>
```

```
<% @bookz.each do |bk|
```

```
  @id = bk.id
```

```
@name = bk.name
```

```
@descp = bk.description
```

```
@price = bk.price %>
```

```
<tr>
```

```
<td> <%= @id %></td>
```

```
<td><%= @name %> </td>
```

```
<td><%= @descp %></td>
```

```
<td> <%= @price %></td>
```

```
</tr>
```

```
<% end %>
```

```
</table>
```

```
</form>
```

```
</body>
```

```
</html>
```

Create Welcome view file

```
> notepad app\views\main\welcome.rhtml
```

```
<html>
```

```
<title> Welcome template for books </title>
```

```
<body>
```

```
<p> Total number of books = <%= @num_books %> </p>
```

```
<form action = "result" >
```

Web Programming Lab Manual

Enter Searching Element: <input type="text" name="sid" />

<input type=submit value="Search" />

</form>

</body>

</html>

Starting Rails Server

ruby script/server

Executing in the web browser

<http://localhost:3000/main/welcome>